

# Not Just Streaks: Towards Ground Truth for Single Image Deraining (Supplementary Material)

Yunhao Ba<sup>1\*</sup>, Howard Zhang<sup>1\*</sup>, Ethan Yang<sup>1</sup>, Akira Suzuki<sup>1</sup>, Arnold Pfahnl<sup>1</sup>, Chethan Chinder Chandrappa<sup>1</sup>, Celso M. de Melo<sup>2</sup>, Suyu You<sup>2</sup>, Stefano Soatto<sup>1</sup>, Alex Wong<sup>3</sup>, and Achuta Kadambi<sup>1</sup>

<sup>1</sup> University of California, Los Angeles  
{yhba,hwdz15508,eyang657,asuzuki100,ajpfahnl,chinderc}@ucla.edu  
soatto@cs.ucla.edu, achuta@ee.ucla.edu

<sup>2</sup> DEVCOM Army Research Laboratory  
{celso.m.demelo.civ,suya.you.civ}@army.mil

<sup>3</sup> Yale University  
alex.wong@yale.edu

## A Visualization of Previous Deraining Datasets

We illustrate some typical image pairs from various deraining datasets in Fig. A. Synthetic datasets in the community are usually generated by adding synthetic rain effects on real images taken under sunny illumination conditions, and the semi-real SPA-Data [34] only considers rain streaks. As a result, the domain gap between these existing datasets and real rainy scenarios are relatively larger as compared with the proposed GT-RAIN dataset.

## B More Results from GT-RAIN

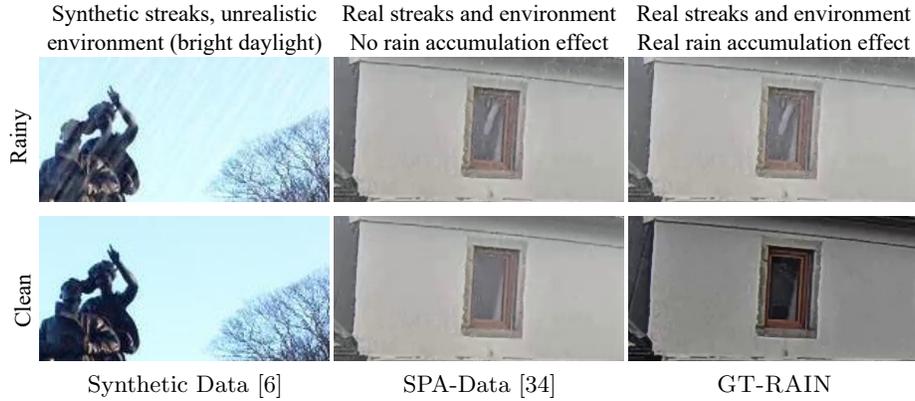
As an additional supplement to Fig. 5 in the main paper, we provide some more quantitative and qualitative results from our test set in Fig. B. Note that these comparison models are using the weights provided by the authors which are trained on synthetic or semi-real datasets. We see that our proposed model trained on GT-RAIN continues to outperform other competing models.

## C More Results on Internet Images

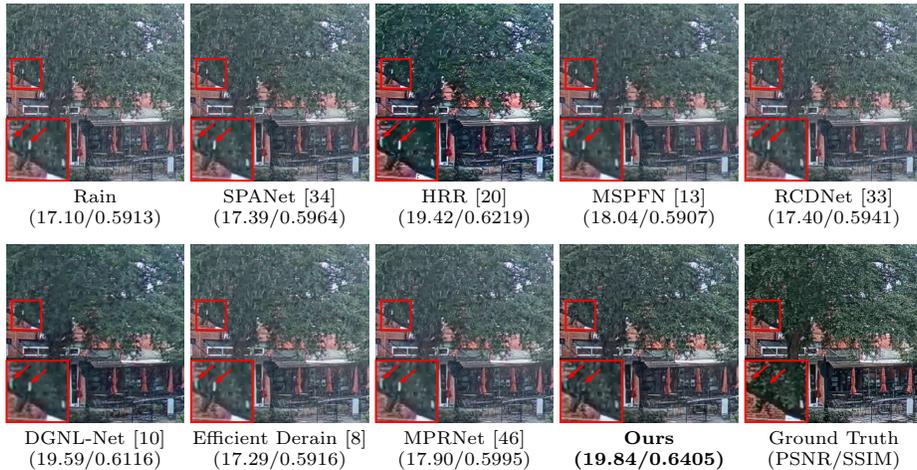
As a supplement to Fig. 6 in the main paper, we provide more qualitative results on real Internet images in Fig. C. Note that all comparison models are using the weights provided by the author, which are trained on synthetic or semi-real datasets. All images are taken from the dataset of common real rainy images provided by [36]. Our proposed model trained on GT-RAIN continues to remove rain streaks of varying shapes and sizes as well as rain accumulation without introducing the unwanted color shifts seen in HRR [20] and DGNL-Net [10].

---

\* Equal contribution.



**Fig. A. GT-RAIN contains realistic rain effects (both rain streaks and rain accumulation), while the existing synthetic and semi-real datasets fail to cover the physical complexity and diversity of real-world rain.** The synthetic image pair is from the commonly used Rain14000 dataset [6], and the pseudo ground-truth image of SPA-Data [34] in the figure is generated by running the official code from the authors on our collected rainy video.



**Fig. B. More results on GT-RAIN test set.** Similarly, the proposed method is capable of removing various rain streaks and rain accumulation effects.

## D Qualitative Results of Retrained Methods

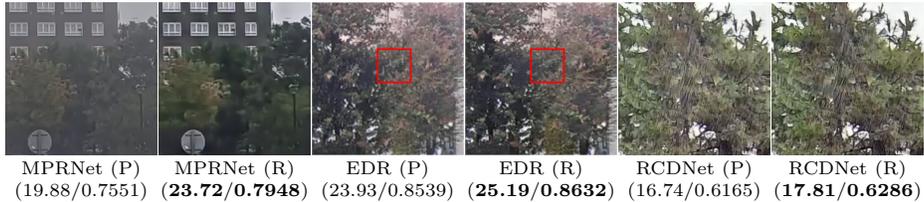
As an additional supplement to Tab. 3 in the main paper, we provide some representative samples of the retrained models for some qualitative comparison in Fig. D. The visual improvements of these derainers in rain fog and streak removal further validate the effectiveness of the proposed dataset.



**Fig. C. More qualitative results on Internet images.** Our model continues to exhibit robust generalization to real rainy images, whereas existing derainers usually fail on removing rain streaks of diverse shapes and sizes. EDR V4 (S) [8] denotes the EDR model trained on SPA-Data [34], and EDR V4 (R) [8] denotes the EDR model trained on Rain14000 [6].

## E Comparison with Semi-supervised Methods

In addition to the models trained on synthetic and semi-real datasets, we also compare the proposed method with some recent semi-supervised methods, including SIRR [36] and MOSS [11], that are trained on real images as a complement to Tab. 2 in the main paper. The corresponding PSNR/SSIM scores on the entire GT-RAIN test set for these two semi-supervised methods are listed



**Fig. D. Qualitative results of retrained SOTAs.** (P) denote pretrained models provided by the original authors, and (R) denotes the retrained models on the proposed GT-RAIN dataset. The improvements further highlight the effectiveness of the proposed GT-RAIN dataset.

as follows: SIRR [36] (20.57/0.6448), and MOSS [11] (21.42/0.7073), where ours are (22.53/0.7304). Some qualitative results can be found in Fig. E.

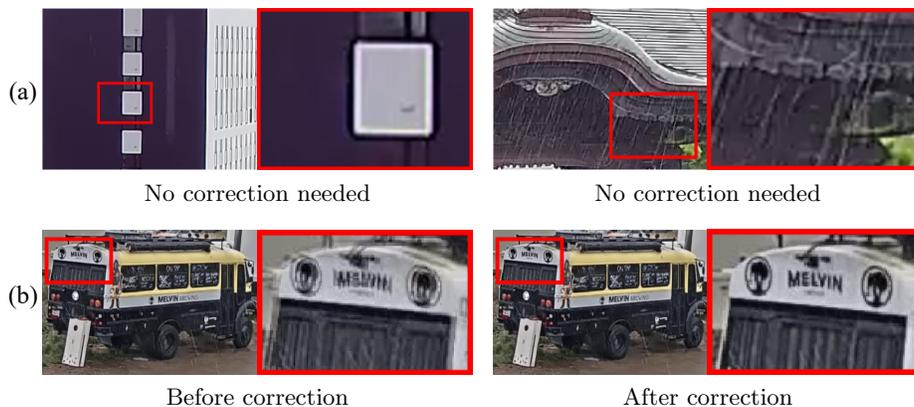


**Fig. E. Qualitative comparison with semi-supervised SOTAs.** As compared with semi-supervised models, the proposed method can remove the rain streaks more effectively.

## F Alignment of Small Motions

As a complement to Sec. 3 of the main paper, we first show, in Fig. F-(a), a ground-truth image overlaid on top of a rainy image to demonstrate representative samples that passed our data collection appearance criteria and also motion criterion, where we do not need to perform motion correction. We note that this is the case for the majority of our dataset. Additionally, we show an overlaid image pair that passed our appearance criteria, but failed the motion criterion. Fig. F-(b) shows the image pair before and after the motion correction. It should be noted that only a small portion of the data requires such correction, and our correction pipeline is designed to be robust to rain artifacts. It is because even though rain can influence local descriptors, the combinatorial matching stage is designed to be robust to a preponderance of outliers. For most

cases, the percentage of outliers affects the time it takes to converge, but not the quality. All samples that require our correction procedure were manually inspected after the alignment – any failure cases of the procedure, typically due to extreme weather conditions, were manually removed.



**Fig. F.** The proposed method can correct for small motions under rain. We illustrate two types of scenes by overlaying the rainy images on top of their clean ground truths: (a) two scenes that do not need additional image processing for motion alignment; and (b) a scene with motion before and after running the correction algorithms. It should be noted that both types of scenes are aligned properly in GT-RAIN.

## G Runtime Comparison

We list the total number of parameters with the associated runtime for other state-of-the-art methods and our proposed model in Table A. The comparison is conducted on a single NVIDIA P100 GPU, and each derainer is asked to restore a colored rainy image of size  $256 \times 256$ . We note that the top three methods (DGNet [10], EDR [8], and our proposed method) all operate at real-time deraining speeds. However, our method outperforms them by 3.73 dB and 2.72 dB PSNR respectively.

## H Limitations

Although we achieve the state of the art for deraining real images, our method is not perfect. Our PSNR and SSIM scores on GT-RAIN are 22.53 dB and 0.7304. This suggests that indeed, we still have ample room for improvement. For example, we leave a slight rain accumulation in the tree in Fig. B. While the recovered image is sharper and contains less rain artifacts than competing methods,

**Table A. Runtime comparison.** The average inference time is calculated on  $256 \times 256$  color images.

Model	SPANet [34] (CVPR'19)	HRR [20] (CVPR'19)	MSPFN [13] (CVPR'20)	RCDNet [33] (CVPR'20)	DGNL-Net [10] (IEEE TIP'21)	EDR [8] (AAAI'21)	MPRNet [46] (CVPR'21)	Ours
Number of Parameters	284k	40.6M	15.8M	3.16M	4.03M	27.3M	3.63M	12.9M
Inference Time (ms)	86.65	35.35	145.5	189.6	4.230	4.617	36.91	12.79

boundaries in highly textured areas (e.g. leaves, bricks, and foliage) are blurred. In Fig. C, we observe a similar trend. However, this is a challenge that plagues all methods. We hope that further extensions of our approach and GT-RAIN will help mitigate these artifacts. We also do not consider occlusions from raindrops on the camera lens because the raindrops will likewise be present on the lens after the rain stops. Moreover, we do not consider specular reflections from water surfaces. This is because these reflections are nearly impossible to reconstruct as the water ripples in the puddles will destroy the visual patterns during raining. We hope that future works can address these limitations. While we have describe image restoration as the main task of deraining, we conjecture that our results may also be applicable towards the re-use of pretrained models on clean data for downstream tasks like: depth completion [9,21,24,25,38,39,40,43,45], stereo [2,3,4,41,44], optical flow [1,17,18,19,31,32], object detection [14,15,22,30], and monocular depth prediction [5,7,27,28,29,35,37,42].

## I Comparison Code Links

The code links for all the comparison methods in the main paper are listed in Table B.

**Table B. Code links for the comparison methods.**

Methods	Links
SPANet [34] (CVPR'19)	<a href="https://github.com/stevewongv/SPANet">https://github.com/stevewongv/SPANet</a>
HRR [20] (CVPR'19)	<a href="https://github.com/liruoteng/HeavyRainRemoval">https://github.com/liruoteng/HeavyRainRemoval</a>
MSPFN [13] (CVPR'20)	<a href="https://github.com/kuijiang0802/MSPFN">https://github.com/kuijiang0802/MSPFN</a>
RCDNet [33] (CVPR'20)	<a href="https://github.com/hongwang01/RCDNet">https://github.com/hongwang01/RCDNet</a>
DGNL-Net [10] (IEEE TIP'21)	<a href="https://github.com/xw-hu/DGNL-Net">https://github.com/xw-hu/DGNL-Net</a>
Efficient Derain [8] (AAAI'21)	<a href="https://github.com/tsingqguo/efficientderain">https://github.com/tsingqguo/efficientderain</a>
MPRNet [46] (CVPR'21)	<a href="https://github.com/swz30/MPRNet">https://github.com/swz30/MPRNet</a>

## J Network Architecture & Implementation

As an additional supplement of the network architecture & implementation section in the main paper, we provide more implementation details here. In our model, the input convolutional block contains two convolutional layers with kernel sizes of  $7 \times 7$  and  $3 \times 3$  respectively. The downsampling blocks are instantiated by  $3 \times 3$  convolutional layers with a stride of 2, and each upsampling block consists of a bilinear interpolation layer and a  $3 \times 3$  convolutional layer. Please refer to Table C for a more detailed illustration of the network architecture. We use batch normalization [12] and choose leaky ReLUs [23] with a negative slope of 0.1 as the activation function. Our model is implemented in PyTorch [26]. The MS-SSIM loss is implemented based on the PyTorch Image Quality (PIQ) library [16]. Experiments are conducted on an NVIDIA Tesla P100 GPU.

**Table C. Illustration of our network architecture.**

Network	Kernel		Channels		Resolution		Parameters	Input
	Size	Stride	In	Out	In	Out		
<i>Encoder</i>								
InputConv1	7	1	3	64	1	1	$\approx 9.5\text{k}$	Rainy Image
InputConv2	3	1	64	64	1	1	$\approx 37.0\text{k}$	InputConv1
DownConv1	3	2	64	128	1	1/2	$\approx 74.0\text{k}$	InputConv2
DownConv2	3	2	128	256	1/2	1/4	$\approx 295.4\text{k}$	DownConv1
DeformResBlock1								
DeformConv11	3	1	256	256	1/4	1/4	$\approx 652.6\text{k}$	DownConv2
DeformConv12	3	1	256	256	1/4	1/4	$\approx 652.6\text{k}$	DeformConv11
Sum1	-	-	256	256	1/4	1/4	DownConv2 + DeformConv12	
DeformResBlock2								
DeformConv21	3	1	256	256	1/4	1/4	$\approx 652.6\text{k}$	Sum1
DeformConv22	3	1	256	256	1/4	1/4	$\approx 652.6\text{k}$	DeformConv21
Sum2	-	-	256	256	1/4	1/4	Sum1 + DeformConv21	
⋮								
DeformResBlock9								
DeformConv91	3	1	256	256	1/4	1/4	$\approx 652.6\text{k}$	Sum8
DeformConv92	3	1	256	256	1/4	1/4	$\approx 652.6\text{k}$	DeformConv91
Sum9	-	-	256	256	1/4	1/4	Sum8 + DeformConv92	
<i>Decoder</i>								
UpConvBlock1								
Bilinear1	-	-	256	256	1/4	1/2	-	Sum9
Conv11	3	1	256	128	1/2	1/2	$\approx 295.2\text{k}$	Bilinear2
Concat1	-	-	128 + 128	256	1/2	1/2	DownConv1, Conv11	
Conv12	3	1	256	128	1/2	1/2	$\approx 295.2\text{k}$	Concat1
UpConvBlock2								
Bilinear2	-	-	128	128	1/2	1	-	Conv12
Conv21	3	1	128	64	1	1	$\approx 73.9\text{k}$	Bilinear2
Concat2	-	-	64 + 64	128	1	1	InputConv2, Conv21	
Conv22	3	1	128	64	1	1	$\approx 73.9\text{k}$	Concat2
OutputConv	3	1	64	3	1	1	$\approx 1.7\text{k}$	Conv22
Total Parameters $\approx 12.9\text{M}$								

## References

1. Aleotti, F., Poggi, M., Tosi, F., Mattocchia, S.: Learning end-to-end scene flow by distilling single tasks knowledge. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10435–10442 (2020)
2. Berger, Z., Agrawal, P., Liu, T.Y., Soatto, S., Wong, A.: Stereoscopic universal perturbations across different architectures and datasets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15180–15190 (2022)
3. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5410–5418 (2018)
4. Duggal, S., Wang, S., Ma, W.C., Hu, R., Urtasun, R.: Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4384–4393 (2019)
5. Fei, X., Wong, A., Soatto, S.: Geo-supervised visual depth prediction. *IEEE Robotics and Automation Letters* 4(2), 1661–1668 (2019)
6. Fu, X., Huang, J., Zeng, D., Huang, Y., Ding, X., Paisley, J.: Removing rain from single images via a deep detail network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3855–3863 (2017)
7. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
8. Guo, Q., Sun, J., Juefei-Xu, F., Ma, L., Xie, X., Feng, W., Liu, Y., Zhao, J.: Efficientderain: Learning pixel-wise dilation filtering for high-efficiency single-image deraining. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1487–1495 (2021)
9. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Penet: Towards precise and efficient image guided depth completion. *arXiv preprint arXiv:2103.00783* (2021)
10. Hu, X., Zhu, L., Wang, T., Fu, C.W., Heng, P.A.: Single-image real-time rain removal based on depth-guided non-local features. *IEEE Transactions on Image Processing* 30, 1759–1770 (2021)
11. Huang, H., Yu, A., He, R.: Memory oriented transfer learning for semi-supervised image deraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7732–7741 (2021)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
13. Jiang, K., Wang, Z., Yi, P., Chen, C., Huang, B., Luo, Y., Ma, J., Jiang, J.: Multi-scale progressive fusion network for single image deraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8346–8355 (2020)
14. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., TaoXie, Fang, J., imyhxy, Michael, K., Lorna, V, A., Montes, D., Nadar, J., Laughing, tkianai, yxNONG, Skalski, P., Wang, Z., Hogan, A., Fati, C., Mamma, L., AlexWang1900, Patel, D., Yiwei, D., You, F., Hajek, J., Diaconu, L., Minh, M.T.: ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference (Feb 2022). <https://doi.org/10.5281/zenodo.6222936>  
<https://doi.org/10.5281/zenodo.6222936>

15. Kalra, A., Stoppi, G., Brown, B., Agarwal, R., Kadambi, A.: Towards rotation invariance in object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3530–3540 (2021)
16. Kastyulin, S., Zakirov, D., Prokopenko, D.: PyTorch Image Quality: Metrics and measure for image quality assessment (2019), <https://github.com/photosynthesis-team/piq>, open-source software available at <https://github.com/photosynthesis-team/piq>
17. Lao, D., Sundaramoorthi, G.: Minimum delay moving object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4250–4259 (2017)
18. Lao, D., Sundaramoorthi, G.: Extending layered models to 3d motion. In: Proceedings of the European conference on computer vision (ECCV). pp. 435–451 (2018)
19. Lao, D., Sundaramoorthi, G.: Minimum delay object detection from video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5097–5106 (2019)
20. Li, R., Cheong, L.F., Tan, R.T.: Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1633–1642 (2019)
21. Liu, T.Y., Agrawal, P., Chen, A., Hong, B.W., Wong, A.: Monitored distillation for positive congruent depth completion. arXiv preprint arXiv:2203.16034 (2022)
22. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
23. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: in ICML Workshop on Deep Learning for Audio, Speech and Language Processing (2013)
24. Merrill, N., Geneva, P., Huang, G.: Robust monocular visual-inertial depth completion for embedded systems. In: International Conference on Robotics and Automation (ICRA). IEEE (2021)
25. Park, J., Joo, K., Hu, Z., Liu, C.K., Kweon, I.S.: Non-local spatial propagation network for depth completion. In: European Conference on Computer Vision, ECCV 2020. European Conference on Computer Vision (2020)
26. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
27. Poggi, M., Aleotti, F., Tosi, F., Mattoccia, S.: On the uncertainty of self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3227–3237 (2020)
28. Poggi, M., Tosi, F., Aleotti, F., Mattoccia, S.: Real-time self-supervised monocular depth estimation without gpu. IEEE Transactions on Intelligent Transportation Systems (2022)
29. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12179–12188 (2021)

30. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
31. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
32. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
33. Wang, H., Xie, Q., Zhao, Q., Meng, D.: A model-driven deep neural network for single image rain removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (June 2020)
34. Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., Lau, R.W.: Spatial attentive single-image deraining with a high quality real rain dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12270–12279 (2019)
35. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2162–2171 (2019)
36. Wei, W., Meng, D., Zhao, Q., Xu, Z., Wu, Y.: Semi-supervised transfer learning for image rain removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3877–3886 (2019)
37. Wong, A., Cicek, S., Soatto, S.: Targeted adversarial perturbations for monocular depth prediction. *Advances in Neural Information Processing Systems* **33** (2020)
38. Wong, A., Cicek, S., Soatto, S.: Learning topology from synthetic data for unsupervised depth completion. *IEEE Robotics and Automation Letters* **6**(2), 1495–1502 (2021)
39. Wong, A., Fei, X., Hong, B.W., Soatto, S.: An adaptive framework for learning unsupervised depth completion. *IEEE Robotics and Automation Letters* **6**(2), 3120–3127 (2021)
40. Wong, A., Fei, X., Tsuei, S., Soatto, S.: Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters* (2020)
41. Wong, A., Mundhra, M., Soatto, S.: Stereopagnosia: Fooling stereo networks with adversarial perturbations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 2879–2888 (2021)
42. Wong, A., Soatto, S.: Bilateral cyclic constraint and adaptive regularization for unsupervised monocular depth prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5644–5653 (2019)
43. Wong, A., Soatto, S.: Unsupervised depth completion with calibrated backprojection layers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12747–12756 (2021)
44. Xu, H., Zhang, J.: Aanet: Adaptive aggregation network for efficient stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1959–1968 (2020)
45. Yang, Y., Wong, A., Soatto, S.: Dense depth posterior (ddp) from single image and sparse range. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3353–3362 (2019)
46. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14821–14831 (2021)